AFRL-RI-RS-TR-2015-053

REALISTIC MODELING OF WIRELESS NETWORK ENVIRONMENTS

CARNEGIE MELLON UNIVERSITY

MARCH 2015

FINAL TECHNICAL REPORT

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND**　　　■　**UNITED STATES AIR FORCE**　　　■　**ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2015-053   HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

     / S /   **/ S /**

MICHAEL T. MUCCIO   MARK H. LINDERMAN
Work Unit Manager   Technical Advisor, Computing
                                              & Communications Division
                                             Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MARCH 2015	FINAL TECHNICAL REPORT	AUG 2010 – SEP 2014

4. TITLE AND SUBTITLE

REALISTIC MODELING OF WIRELESS NETWORK ENVIRONMENTS

5a. CONTRACT NUMBER
FA8750-10-1-0232

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
N/A

6. AUTHOR(S)

Peter Steenkiste

5d. PROJECT NUMBER
RELM

5e. TASK NUMBER
OD

5f. WORK UNIT NUMBER
GC

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITF
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER

AFRL-RI-RS-TR-2015-053

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Wireless networking has become a part of our everyday life, but it is a challenging networking technology. The reason is that wireless channel properties, and thus link quality, depend strongly on the physical environment. This creates a challenge for both simulation and emulation platforms, since they must accurately model the impact of many physical layer factors on signal propagation in order to achieve a high degree of realism. This project developed a number of techniques to improve the accuracy of developing realistic and efficient environment-specific channel models. We looked at how the accuracy of both geometric and stochastic channel models can be improved by generating model inputs that accurately reflect a specific environment. Our evaluation uses two important properties, channel dynamics and correlation between nearby channels, in a challenging wireless environment, namely vehicular networks. We also made a number of improvements to an emulation-based wireless testbed to improve channel model accuracy and transferred a copy of the testbed to LTS as the basis for research collaboration.

15. SUBJECT TERMS

Wireless, Networks, Modeling, Simulation, Emulation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON **MICHAEL T. MUCCIO**
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	UU	33	19b. TELEPHONE NUMBER *(Include area code)* **N/A**

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Table of Content

Table of Figures

1.0 SUMMARY

Wireless networking has become a part of our everyday life, but it is a very challenging networking technology. The reason is that wireless channel properties, and thus link quality, depend strongly on the details of the physical environment. This creates a significant challenge for both simulation and emulation platforms, since they must accurately model the impact of various physical layer factors on signal propagation in order to achieve a high degree of realism. Moreover, network-level simulation involves many channels so efficiency becomes a major concern. Today's wireless protocols often adapt to channel conditions so it is not only important to model individual channels accurately, but we must also consider correlation across nearby channels.

This project developed a number of techniques to improve the accuracy and efficiency of environment-specific channel models. First we added a number of features to an emulation-based wireless testbed to improve the accuracy of wireless channel modeling and to expand the type of experiments that can be supported. Improvements include better A/D and D/A convertors, improved packaging and shielding, and a significantly faster control channel that supports more frequent updates of the channel state. We also added large memories to the SCM and DSP card, allowing us to accurately model interference from various types of devices. Finally, we added a front-end that can be tuned across a large spectrum range, opening the door for DSA experiments.

Next, we looked at how the accuracy of both geometric and stochastic channel models can be improved by generating model inputs that accurately reflect a specific environment. We developed a technique that uses aerial photographs to estimate the density of objects in an outdoor environment, which can then be used as inputs for a geometric fading model. We show that using this technique to generate location-specific inputs cuts the modeling error relative to ground truth in half, compared with using average values as inputs. Unfortunately, this technique is quite expensive. This suggests that a hybrid approach is needed. Use geometric models with accurate inputs based on measurements for the most significant channel features (e.g., fading and LOS blocking by large objects, but stochastic models may have to be used for features that are driven by more fine grain properties of the environment (e.g., impact of small objects).

We also look at the problem of modeling spatial correlation between links in the same physical environment, a feature that is often ignored in simulators and emulators. We compare the cost and accuracy of adding spatial correlation to both geometric and stochastic models. Geometric models that use inputs consistently automatically capture spatial correlation, but for stochastic models, an explicit spatial correlation model must be added. While possible, this model becomes expensive as the number of correlated channels increases, so it is not practical for dense wireless networks, or for channel features with a large geographic scope.

2.0 INTRODUCTION

Wireless networking has become a part of our everyday life, but it is a very challenging networking technology. In contrast to wired networking technologies, the complex interactions between signal propagation and the physical environments result in very dynamic physical layer properties that fundamentally affect all layers of the protocol stack. As a result, the evaluation of wireless networking technologies is very challenging since all aspects of the system must be considered, including the physical environment and the entire protocol stack. Because of these challenges, a wide range of evaluation technologies have been developed, offering different tradeoffs between level of realism, repeatability of experiments, ease of use, and scalability. Examples include anechoic chambers, various types of testbeds, simulators, and emulators.

The goal of this proposal is to develop techniques to improve the accuracy of two widely used wireless network experimentation platforms, namely simulation and emulation. The two techniques are quite different. Simulation must recreate the wireless devices, which often requires significant simplification [1,2]; its primary benefit is scalability. Emulation, on the other hand, uses real devices including real radios, protocol stacks, and applications, but its scale tends to be limited to a few tens of devices. Despite these differences, both techniques rely on channel models to recreate the signal propagation environment. In fact, the use of channel models is what gives both platforms their flexibility, allowing them to be used for very diverse experiments. Of course, the value of simulation and emulation-based tools depends critically on the accuracy of those channel models.

Many statistical channel models have been developed and validated. However, they are generic and do not capture the effects of a specific physical environment. Examples of relevant physical factors that impact signal propagation include line of sight (LOS), multi-path, mobility, correlation between the changes in nearby channels, and interference from external RF sources. Some simulations consider mobility, e.g. the random waypoint model, but this is a very crude approximation and the other effects are often ignored. Unfortunately, these macroscopic effects significantly affect how the SINR and other properties of the received signals changes over time. This in turn has a large impact on the behavior and performance of many wireless protocols (e.g. rate adaptation, routing, etc.), in ways that are environment specific. This means that we need to model how the physical environment affects signal propagation in order to accurately evaluate wireless network technologies in various environments.

To address these shortcomings, we developed a set of "world models" that model these macroscopic, physical world effects and their impact on signal propagation. We implemented these world models for a wireless emulator testbed and we extended them to model the effects of external interfering RF devices. While we focused on the development of world models for an emulation-based wireless testbed, the techniques also apply to simulators. As part of this project, we will also help build a copy of the CMU wireless emulator testbed at LTS as the basis for research collaboration.
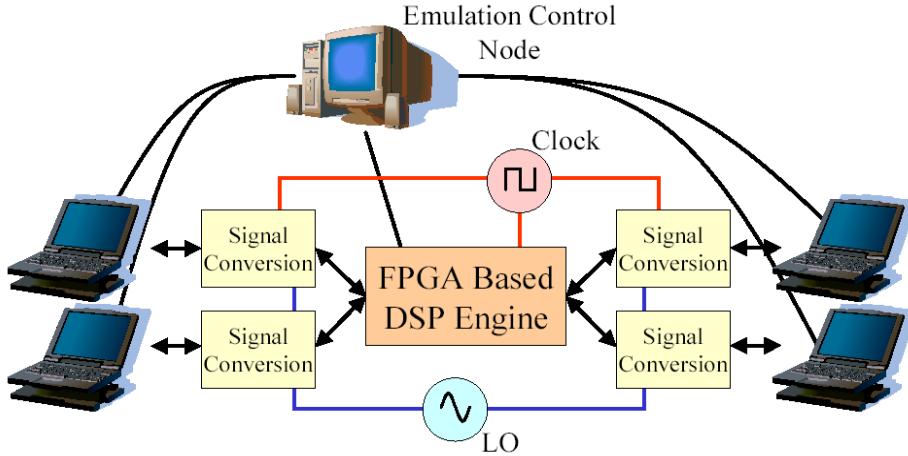
Figure 1: Wireless Emulator Testbed Architecture

3.0 BUILDING A FLEXIBLE EMULATION-BASED WIRELESS TESTBED

We present a high level overview of the emulator architecture, discuss the features that were added to the emulator design to meet LTS requirements, and describe the system that was built.

3.1 Emulator Overview

We have developed a wireless networking testbed based on signal propagation emulation [3,4]. The operation of the CMU wireless emulator is illustrated in Figure 1. A Signal Conversion Module (SCM) and radio frequency Front End (FE) take the signal transmitted by a wireless device from its antenna port, convert it to the digital domain, and forward it to a Digital Signal Processing (DSP) card that uses an FPGA to emulate the effect of signal propagation (e.g., attenuation, multi-path, and small-scale fading, interference). The processed signal is then sent back to the wireless devices through the SCM and FE [5]. The emulator simultaneously offers a high degree of realism and control. The devices are shielded from each other so that no communication occurs over the air. Since the devices only communicate through the emulator, we have full control over signal propagation. Experiments are controlled using software running on the emulation control node [6].

A first version of wireless emulator testbed was in regular use by both CMU and external users from 2007 until 2011 [7,8]. It supported the full 2.4 GHz ISM band and had 15 nodes. The nodes were laptops with 802.11b/g interfaces based on an Atheros chipset. A number of experiments also used software-defined radios (USRP with GNU Radio) and Bluetooth devices. The experiments that used the emulator are very diverse. Classes of experiments include the characterization of wireless links [9,10] and devices [11], analyzing real world wireless behavior [12], comparing and evaluating wireless protocols [13,14,15], physical layer experiments [16], and course projects and assignments.

In 2009, we received funding from NSF to build a new version of the emulator. The primary goal was to improve the fidelity of channel emulation. A secondary goal was to improve the packaging of the system and the shielding of the nodes. As part of this project, we identified the following improvements to the emulator SCM and DSP cards:

- We upgraded the FPGA SCM to have more compute cycles. This will allow us to do more elaborate pre-processing of the signals before they are passed on the central DSP card.
- We replaced the A/D and D/A converters with converters that support IQ sampling in hardware. The new D/A converter also has higher precision.
- We added a second set of A/D and D/A converters to the card so the cards can support two wireless devices with a lower spectral bandwidth, instead of one high bandwidth node. Similar to the old design, the card can support one node using 100 MHz of spectrum, or two nodes using, for example, 40 MHz of spectrum.
- We replaced the FPGA on the DSP card with a more powerful model, allowing us to model many more RF paths per channel.

3.2 Methods, Assumptions, and Procedures

In a meeting with LTS, we had a design review of the above design. We also discussed additional features that would add value to the emulator testbed without requiring radical design changes. We then followed up with a sequence of conference call to discuss what additional features to include in the design. Based on these discussions, and information collected by LTS about what type of features were must critical internally, we decided to make two following additional design changes:

- We added memory to the SCM card that is accessible from the FPGA. The memory can be used for a number of tasks, including capturing samples, storing samples for replay, and storing parameters for channel models. We also increased the size of the memory available on the DSP card so longer traces can be stored and replayed.
- We replaced the parallel copper cables that connect the SCM to the central DSP card with optical fiber. This has three advantages. First, it simplifies packaging since the copper cables are thick and hard to work with. Second, it improves isolation of the signals, and thus fidelity. Third, the use of a self-clocked serial link improves reliability.
- We added a control interface so the SCM can be controlled directly from the emulation control node. SCM control in the old emulator testbed needs to flow through the central DSP card, which adds complexity to the code.
- We decided to replace the old front-end, which only supported the 2.4 GHZ ISM band, with a wide band front-end that can support a wide range of frequency bands of interest to both LTS and CMU. A highly desirable feature that was identified was the ability to switch quickly between bands in order to support experiments in dynamic spectrum access.

The detailed design and the layout of the SCM and DSP boards was outsourced to outside companies (DRG Engineering and AmeriCAD, respectively). Designs were reviewed by both CMU and LTS researchers. The building of the cards (manufacturing

and assembly) and basic testing was also outsourced (Hallmark Circuits and Artek Manufacturing). Functional testing was done by CMU, in collaboration with LTS. For all cards, we initially built a small number for testing purposes. Only after those test cards passed functional tests of all major functions did we replicate the card. We discovered minor problems with both the initial SCM and DSP card and these were addressed before the cards were replicated.

The Xilinx code on both the SCM and DSP cards required significant changes relative to the code that was used in the old emulator testbed. Changes were needed both for testing purposes and for use of the cards in a production environment. The changes include adding support for new functional blocks (optical interfaces, memory access, new control interfaces) and for upgraded version of other functional blocks (fast Ethernet interface, A/D and D/A). The upgrades of the FPGA also allowed us to have an embedded core on the FPGAs, which simplifies the control of the cards.
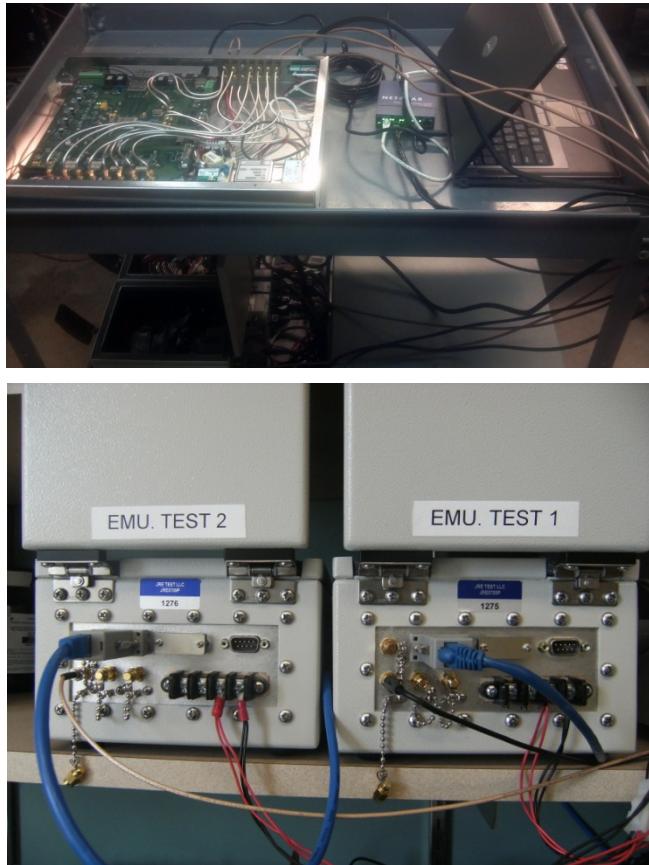
Figure 2: SCM-based Emulator (top) Connecting Two Nodes (bottom)

In order to validate both the new SCM hardware and software, we used the new SCM card to build a small, two node emulator testbed in Spring 2013. Figure 2 shows the setup, including both the SCM (top) and the two wireless devices used in the experiment (bottom). This testbed was used for point-point vehicular wireless experiments that used the world model described in Section 4.

The front-ends required a complete new design. It is an analog card that is similar to the front-end of dynamic spectrum access (DSA) radio. The primary difference is that, compared with a DSA front-end, the emulator front-end also needs a mechanisms to switch between transmit and receive modes in order to accommodate full-duplex radios; switching must be controlled by a spectrum sensor that detects the presence or absence of a signal from the attached radio. We selected Radio Technology Systems to design the cards because they have long track record in this area. The design strategy used was similar to that of the SCM and DSP cards, except that it involved more extensive interactions between Radio Technology Systems, CMU, and LTS because of the much more aggressive nature of the card. PCB fabrication and assembly was outsourced to Imagineering. The goal was to design a card that can fairly quickly switch (well under 1 msec) between channels across a wide range of frequencies, from as low as the TV white spaces (~500 MHz) to as high as the frequency band allocated for DSRC-based vehicular networks (~5.9 GHz). We were able to meet that goal, as described below.
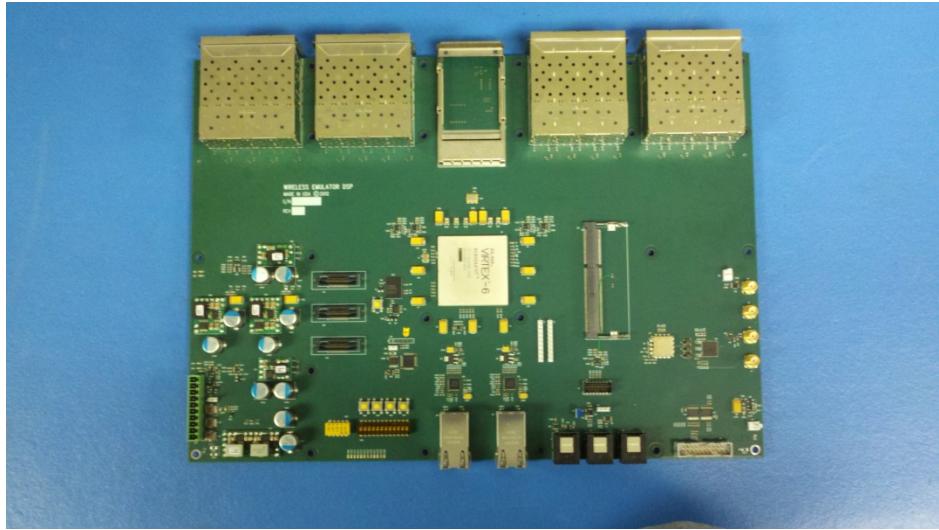
Figure 3: DSP Card

3.3 System design and implementation

The high level specifications for the emulator system are:

- An analog channel bandwidth of 100 MHz
- A sampling rate of at least 100 MHz (complex), or 200 Msamples/second total
- A sample precision of 16 bits
- Supports up to 16 ports for a total of 16 * 15 = 240 channels

The hardware also allows connecting up to 32 ports at a lower channel bandwidth, e.g., 50 MHz. We now discuss the three custom cards used in the emulator:

3.3.1 Digital signal processor (DSP)

The DSP card (Figure 3) provides centralized computation for all signal propagation and distortion between the emulator's end-points. There are four major hardware subsystems.

First, **computation** is performed by a Xilinx Virtex 6 (XC6VSX475T) FPGA with embedded digital signal processing slices. This FPGA contains 2,016 such slices, each of which can operate at up to 600 MHz. This provides a maximum computational throughput of $1.2 * 10^{12}$ real multiplications per second or $3 * 10^{11}$ complex ones. The FPGA also contains 38 Mb of SRAM in blocks which are tightly coupled to the DSP slices. This memory is structured as just over 2000 blocks, all of which can be written to and read from concurrently. In our example configuration, that means we can do real-time processing of up to 3000 complex channel taps. This allows us to simulate a 12-tap channel between every pair of nodes. For comparison, the previous DSP had only enough resources to give every channel 1 tap, and a few hard-coded channels 3 taps.

Second, in addition to the 38 Mb of on-FPGA storage, the DSP card supports an external DRAM memory module with up to 8 Gb of additional **storage**. Unlike the on-FPGA memory, only a relatively small number of samples (roughly 2 64-bit words, depending on achievable bus rates) can be read or written per clock cycle. This memory therefore does not lend itself to storing samples for propagation delay, but could be used for signal recording and replay (including precisely-controlled interference experiments), buffering channel updates, or other yet to be determined purposes.

Third, **high-speed communication** of the digitized RF samples to and from SCM cards uses dedicated serial links. The exact physical media are modular and replaceable: Both cards support interchangeable SFP (Small Form-factor Pluggable) transceiver modules. With 850nm fiber-optic transceivers, full-duplex line rates of 5 Gb/s are consistently supported. This allows for 312 MSPS (real) or 156 MSPS (complex) per serial link. The DSP card has a bank of 16 such transceiver modules, to support 16 attached SCM devices. In comparison, the parallel electrical connections used in the previous version of the Emulator offered roughly 2 Gb/s of bandwidth to each SCM. This new design more than doubles the throughput, and equally importantly importantly, it improves connection quality: The new serial connection architecture supports clock recovery and de-skewing and robust error correction and detection.

Finally, the DSP supports two **control interfaces**. Since most channel model processing is performed on a dedicated general-purpose computer, we need a high-speed link to send low-level channel-setting commands to the DSP board. As with the SCM links, this link is supported by a modular transceiver unit. We are using a 10 Gb/s Ethernet transceiver currently, supporting roughly 120 M channel changes per second. The DSP card (and all the SCM cards) also includes a gigabit Ethernet transceiver for asynchronous monitoring and configuration messages. This is separate from the (channel) control data path, and could be used for system configuration changes, status monitoring, and loading data into (or reading data from) bulk memory.

3.3.2 Signal conversion module (SCM)

The SCM card (Figure 4) includes three logical blocks: AD/DA conversion, digital interface with the DSP, and a control block.
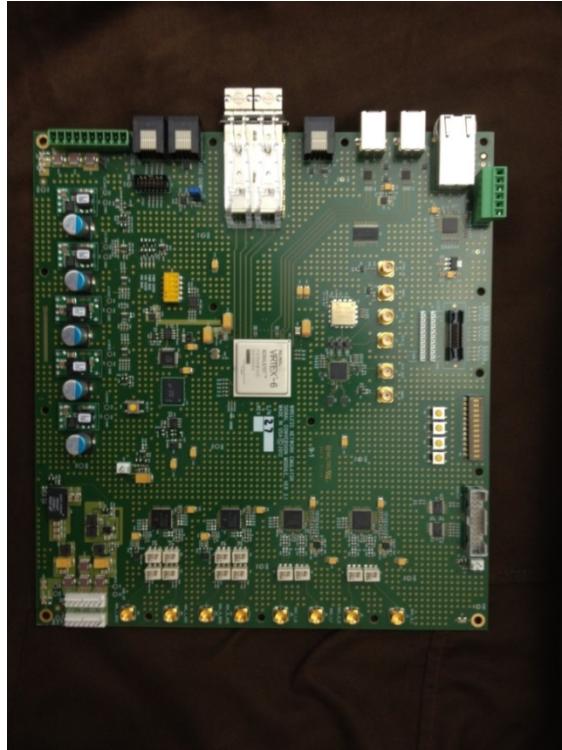
Figure 4: SCM Card

The SCM card has two dual-channel ADC chips, and two dual-channel DAC chips, for a total of 4 inputs and 4 outputs. The inputs and outputs can function as either 4 independent channels (each way) or 2 complex (I/Q) pairs. The RF Front End (discussed below) is designed to work with complex baseband signals, but the SCM itself is agnostic. This design allows an SCM to support a single device with 2-way antenna diversity or MIMO, or two separate concurrent devices. Two SCMs can be ganged together to support 4-antenna devices. The ADC chips sample at 250 MSPS per input (up to 125 MHz analog) with 12 bit resolution. Their quality measures at the expected frequencies include a Signal to Noise Ratio (SNR) of 66 dB and a Spurious-Free Dynamic Range (SFDR) of 80 dB. The DAC chips sample at 500 MSPS per output (up to 250 MHz analog) with a 16 bit resolution. It has a SFDR at the expected frequencies of 70-80 dB. This high sample rate will allow us to oversample for better accuracy.

Each SCM has two main serial transceivers for communicating with the DSP; they are as described in the DSP section.

Each SCM also has a gigabit Ethernet interface for asynchronous control; this is identical to the one described in the DSP section as well. Each SCM card also has one RS-485/UART transceiver for low-speed character communication, and two general-

purpose high-speed serial transceivers using USB3 PHY chips.   There is no current plan to use these interfaces; they are for extensibility. Finally, control and monitoring of the RF front end cards will be done by software running on the SCM card. The SCM includes a general-purpose 20-pin bidirectional interface which will be used for this purpose.

3.3.3 RF Front Ends

The RF Front End (FE) has three major functions:

1. Frequency translation: The SCM modules are designed to process baseband signals in the 0-200 MHz range. The RF Front End translates these signals up to and down from the operational frequency band of the devices under test.
2. Power matching: The AD/DA converters in the SCM have an optimal full-scale voltage (roughly 1 Vpp). To obtain the maximum resolution of the SCM (and avoid distortion or damage), RF signals from and to the devices under test are attenuated and/or amplified to match the desired power range.
3. Half-duplex (time-division duplex) switching: Most devices under test are likely to use a single antenna port for both transmitting and receiving. The RF Front End is responsible for switching its connection to this port between output and input operation.

We now elaborate on these three functions.

Frequency conversion - The RF Front End (RFFE) translates radio frequency ("passband") signals down to, and up from, complex baseband signals. This section will describe the card in terms of down-conversion; the up-conversion capabilities and process are symmetrical.

The SCM expects baseband signals in the form of I/Q pairs in 0-100MHz range (for a total bandwidth of up to 200 MHz). The RFFE supports signals in a continuous RF frequency range of 400 MHz to 6 GHz, which it converts to the range used by the SCM. The transmit and receive chains within a given RFFE card can operate at different frequencies for working with frequency-division duplex (FDD) devices, and every RFFE card within the emulator can operate at a different frequency.

Each RFFE contains two frequency synthesizer circuits which control the center frequencies of the up- and down-conversion process. Either synthesizer can be used for either (or both) up- and down-conversion. There are three mechanisms for frequency selection:

- (Primary) Digitally setting frequency synthesizer parameters. This allows each synthesizer to be set over the entire frequency range in relatively fine steps. In general, this involves a the synthesizer PLL unlocking a re-locking at the new frequency, requiring 50 - 200 microseconds.
- Synthesizer cut-over: If the two synthesizer are not being used concurrently for frequency-division duplexing, fast frequency changes can be accomplished by switching between the two.   Including control communication delay, this switch should require around 300 nanoseconds.

- Analog bias: Both the PLL and on-board reference oscillator can be biased using on-board DACs to make smooth small frequency shifts without discontinuities from breaking PLL lock or switching synthesizers. The range, accuracy, and delay of this approach will have to be determined empirically.

The frequency synthesizer outputs go through filter banks consisting of five banks for different sections of the frequency range. These serve to suppress harmonics and spurs from the synthesizer before they cause distortion in the mixing process. A large frequency change may therefore require a filter switch, adding roughly 100 nanoseconds of latency.

The RFFE also contains extensive filtering of the RF signal to reject unwanted signals. This serves two purposes: First, it ensures that when the RFFE is used in the emulator, out-of-band RF emissions from the device under test (e.g. circuit noise, on-board transceivers other than the one being used) are not aliased into the observed signal. Second, it allows the RFFE to be used for transmitting and receiving over-the-air signals outside the emulator environment, in which case off-channel radio transmissions are the primary unwanted signal source.

Power Matching – In general, the power level of the baseband signal should be roughly constant, while the power at the RF interface can vary extremely widely.

On the baseband side, there are several reasons for this: First, the ADC and DAC devices operate over an essentially fixed voltage range. An over-powered signal going into the ADC will be clipped, producing severe distortion, and the DAC cannot produce an out-of-range output. On the other side, an under-powered signal into the ADC (or out of the DAC) throws away a significant portion of the range (and therefore resolution) of the devices, substantially reducing the accuracy of the emulator. Additionally, within the RF Front End, many analog components give their best performance over a limited voltage range. On the RF side, reasonable device transmitted powers (input to the RFFE/emulator) could be as high as +30 dBm or as low as -70 dBm. Corresponding received powers (output from the RFFE/emulator) would range from -10 dBm down to the noise floor, roughly -110 dBm for 1 MHz channel. Many applications fall further outside this range, but we do not plan on supporting them without external attenuation or amplification.

The RFFE includes a series of digitally-controlled amplifiers and attenuators, including both coarse fixed-gain devices which are switched in and out, and fine variable-gain devices with direct digital control. These accommodate an input power level ranging from +30 dBm to -115 dBm, and produce an (full-scale) output power level ranging from +16 dBm to -115 dBm, with minimal loss of linearity. Additionally, DC power can be coupled onto the RF input and output connectors to power an external amplifier.

In general, it is expected that the expected power ranges will be defined in advance of any particular experiment: e.g. for any common WiFi scenario, the input power would be expected to be in the +0 to +20 dBm range, and the ouptut power in the -40 to -95 dBm range. The RFFE gains can be fixed accordingly, and input the SCM will vary with both the signal itself and transmit power control, while the output from the SCM will vary with the signal(s) and changing channel attenuation.

Duplex Switching – Most devices under test are likely to use a single antenna port for both transmitting and receiving. The RF Front End is responsible for switching its connection to this port between output and input operation.

The RFFE has 4 RF I/O ports, each of which can be connected to either the transmit or receive chain. This allows for switching between attached multiple attached devices and/or calibration and test ports, but the primary purpose is for duplex switching. Switching can be controlled in two ways.

The most common expected case is that the emulator has no prior information about whether an attached device is transmitting or receiving at any given moment. In this case, the emulator will assume that all devices are (potentially) receiving at all times until it detects transmit-level power on a device's RF port. So long as a device's minimum transmitted power exceeds the maximum plausible received power, a detection threshold set between the two will reliably identify transmissions. Almost any realistic environment will include at least 30 dB of attenuation between any pair of antennas, so the margin for setting such a threshold is generally wide. The RFFE provides a programmable detection threshold of approximately -55 dBm to 0 dBm.

Non-constant-envelope modulation schemes (including common ones like OFDM), can have significant power variation during a transmission, meaning that a "high-power" signal could be arbitrarily weak from moment to moment. To handle this, the RFFE provides a programmable hysteresis loop, so that the received power must remain low for a predetermined period before the power-detected bit is cleared. The longer this period is set to, the lower the probability of incorrectly switching to receive mode during a transmission. However, a device will also be unable to receive for that duration after it is actually done transmitting. The hysteresis delay is programmable from 50 nsecs to 24 μsecs, to accommodate experiments with different network technologies. The total switching time should be on the order of 300 nanoseconds in the fastest configuration.

An alternative to switching based on power-detection is explicit switching using the digital control interface. In that case, the control software (running on the SCM board) can use arbitrary logic, but the switching time will be much higher.

Precise timing reference is essential to accurate RF measurements. The RFFE supports three timing sources: One external input, and two on-board oscillators. Using an external reference allows every RFFE in the emulator to be coherent, and synchronized with the SCM and DSP boards. Using an on-board oscillator allows the frequency to be biased, for example to simulate Doppler shift or oscillator drift, without the added latency of a digital filter. They also allow the RFFE to be used for standalone measurement and communication without any external reference.

3.4 Results and Discussions

The SCM, DSP and RFFE cards have been built and are being integrated into a complete emulator testbed both at CMU and at LTS. Figure 5 shows for example the enclosure that holds and emulator "port". It holds an SCM card, RFFE and an attached RF device (SCM shown).

The new emulator hardware meets the requirements identified at the start of the project. First, the hardware support channel models with significantly higher fidelity through the use of better AD/DA papers, bigger FPGAs on both the SCM and DSP cards, higher speed control paths, and better shielding and packaging of the system. Second, it provides support for DSA-experiments through the use of RFFEs that cover a wide spectrum range and fast switching. Finally, it supports a wider range of experiments through the addition of large memories on the SCM and DSP cards, and more direct control of each card.
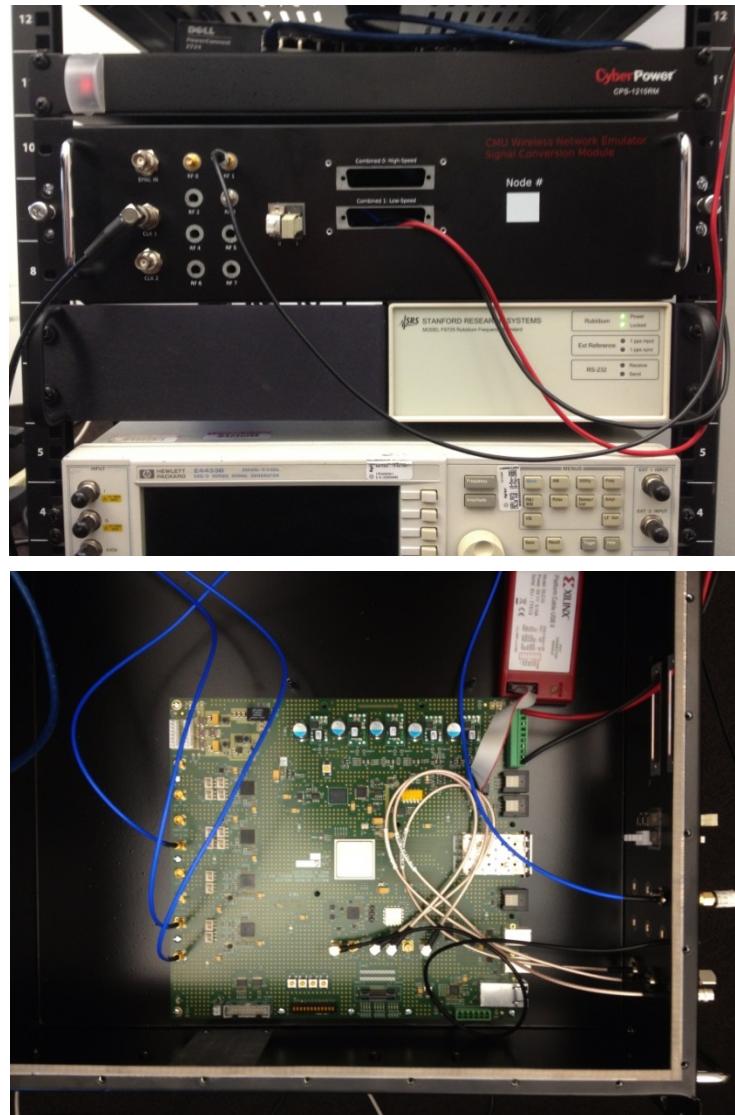
Figure 5: Front and Top view of Port Enclosures (Shows SCM)

4.0 A WORLD MODEL FOR ACCURATE CHANNEL MODELING

We describe the design and implementation of an efficient channel simulation architecture that captures channel dynamics, spatial correlation, and the impact of external interference in an environment-specific fashion.

4.1 Motivation

The physical environment of wireless network influences the properties of wireless channels in complex ways. Moreover, link properties are highly dynamic because the signal propagation environment changes as a result of movement in the environment. Many models for various channel properties such as path loss, multi-path [17] and fading have been developed, capturing the impact of the physical environment. Examples include the log-distance path loss model, Nakagami fading model, and a variety of fading models for specific environments.

The performance of many wireless network protocols and applications is sensitive to changes in wireless channel (e.g., signal strength) or link (e.g., packet delivery rates, bandwidth) properties. In order to optimize performance, protocols and application often adapt to changes in the wireless link properties. Examples include transmit rate adaptation, TCP congestion control, and video bit rate adaptation. Some protocols and applications are not only sensitive to changes in the properties of individual links, but also to how these changes are correlated across nearby links. Examples include routing protocols that adapt to quality of the links in multi-hop wireless networks, e.g., [12], or protocols that rely on opportunistic overhearing of packets, e.g., [13,15]. In order to evaluate these protocols, simulators and emulators not only need to accurately model the channel or link dynamics of individual links, but they also need to properly model the correlation of these properties across nearby links. Finally, since a network-level simulation needs to model a large number of links, as many as N^2 for an N node network, the channel models must be very efficient.

The rest of this section describes how we use a "world model" to efficiently and accurately model various channel properties, capturing channel dynamics, spatial correlation, and the effect of external interferers. Our focus is on support environment specific models, i.e., to support the simulation and emulation of wireless networks in specific environments.

4.2 Methods, Assumptions, and Proceudures

Models for various channel properties can be loosely classified as either geometric or stochastic. Geometric models are based on simple model of how objects in a specific environment affect the property of interest. They take as input parameters that further characterize the physical environment, e.g., distance between devices, speed and density of objects, etc. Stochastic models on the other hand are based on a random process and input parameters are selected in such a way that the output of the model matches the channel properties observed in the target environment. In practice, many models are not "pure" geometric or stochastic models, e.g., geometric models may take

random inputs, or stochastic models may take some inputs that are environment specific. Geometric models tend to be more accurate but they are typically more expensive and obtaining accurate input parameters can be very time consuming.

Given that our primary goal is the accurate simulation and emulation of specific environments, we will focus on improving the accuracy of geometric models of channel properties. The accuracy of the channel modeling depends on two factors. First, we need to determine accurate input parameters for the models, *i.e.*, parameters that reflect the environment of interest, and we need to also capture how these parameters change over time. Second, we need to capture the spatial correlation across channels in the simulated wireless network.
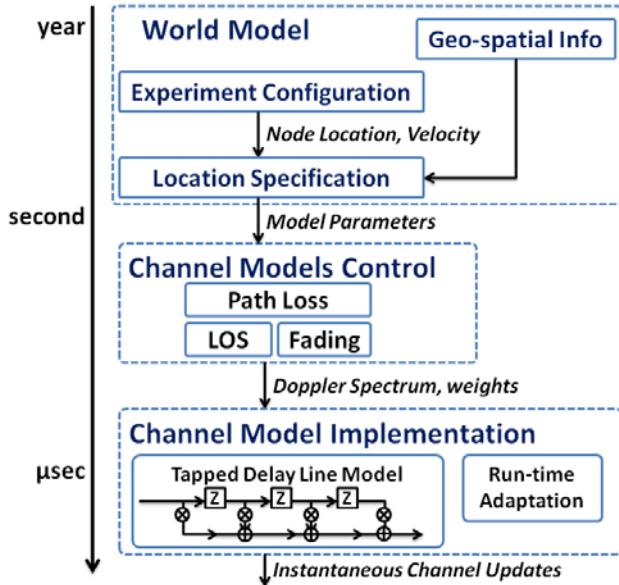
Figure 6: Simulation Framework Using World Model

Our approach is based on the use of a model of the physical environment that represents features that are relevant to the channel properties of interest. This (physical) "world model" must keep track of both properties of the objects in the environment (including the radios) and how these properties change over time. They must also have enough spatial information so it can determine the degree of spatial correlation across links. Note that this degree may depend on the channel property being modeled. For example, the geographic scope of the correlation may be different for LOS properties caused by objects, and loss of transmission opportunities due to radios that are not explicitly simulated.

In order to make our research results more concrete, we decided to focus our evaluation on a specific wireless environment, namely vehicular networks. We picked this environment because it is very challenging both in terms of channel dynamics and spatial correlation, and because we have access to detailed channel traces that we can use as ground truth for the channel dynamics evaluation.

4.3 Capturing the Impact of Physical Effects

We describe our architecture for accurate modeling of channels.

4.3.1 World model frame work

Figure 6 shows the high level architecture for wireless channel simulation and emulation platform. From top to bottom, it includes three major components: A world model, channel model control and channel model implementation, considering time scales from almost static to μs. We now briefly discuss key features of each of the components.

Our channel model implementation uses a standard multi-tap delay line to model multipath effects. In order to be able to efficiently combine the impact of different types of effects on the channel, we work in the frequency domain by calculating the Doppler spectrum for each effects, e.g., LOS, fading and scattering. These are then added using weights. More details can be found in [18]. Both the Doppler spectra and their relative weights are update at runtime to model the dynamics in the environments. They are calculated by the Channel Model Control component.

The Channel Model Control component is responsible for calculating and updating the Doppler spectra corresponding to different physical phenomena impacting the channel. Here, we can often reuse existing channel models, which, as pointed out earlier, are either of a geometric or stochastic nature. Both types of models require some information about the physical environment, which is provided by the world model. To make this more concrete, let us briefly describe a vehicular channel model for an urban environment. In typical urban/suburban vehicular networks, the two primary paths are line of sight (LOS) and reflections off buildings and possibly other objects (e.g. trees) lining the street. Our model considers four physical layer effects [19]:

- A pathloss model: we use the log-distance model.
- A model for fading due to reflections off large objects such as buildings: we use the "cone" model [20], which is a geometric model for urban streets.
- A model for scattering off small objects: we use a simple stochastic model.
- A LOS-model that models loss of LOS due to cars, buildings, etc.: we use a two-state stochastic model.

More information can be found in [19]. We use the cone fading model as our case study for accurate modeling of channel dynamics, and the LOS-model as the case study for modeling spatial correlation.
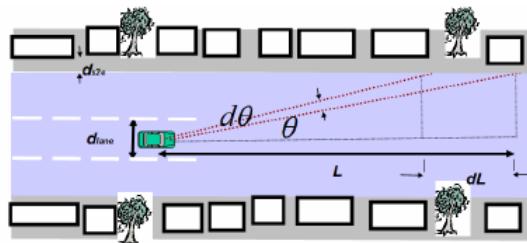
Figure 7: "Cone" Fading Model for Urban Street

4.3.2 Modeling channel dynamics

The accuracy with which a channel is modeled depends on both the model and the input parameters. Let us use the cone model as an example. As shown in Figure 7, it models the vehicles and large objects along the street to calculate the fading spectrum). The parameters it requires include: the transmitter and receiver velocities and their relative distance, the density of roadside objects, the width of the lanes, and the number of lanes separating vehicles from the sides of the road. As shown in Figure 8, these input parameters can be obtained in a number of ways, including artificially generated models, user input, or they can be calculated based on measurements of an actual environment. Since we are interested in environment-specific models, we focus on the third case.
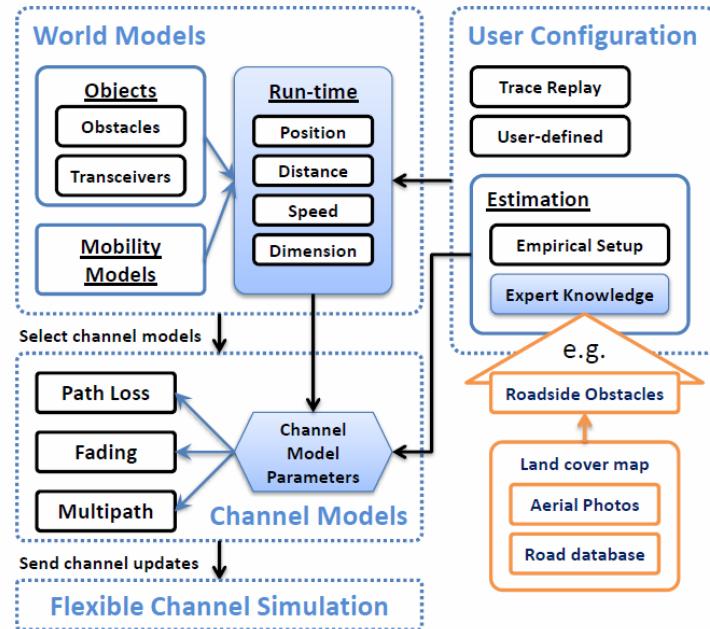
Figure 8: Generating Inputs for Channel Models

Two types of inputs were used to calculate the inputs for the four models used in the vehicular simulation. First we used a set of signal-level measurements of a vehicle-to-vehicle channel collected in Squirrel Hill, an urban neighborhood near the CMU campus [21]. The trace consists of snapshot of the propagation channel at high frequency and they were used to generate Doppler spectra for each snapshot. The trace also provides timestamps and the GPS location of the two vehicles. This trace was used to generate input parameters for three of the models:

- Distance between vehicles for the path loss model.
- The trace allowed us to determine when there was LOS between the two parameters. These statistics were used to generate inputs for the stochastic LOS model.
- We did not have information on the small scatterers in the environment. The impact of small scatterers shows up as small "peaks" in the Doppler spectrum.

We used statistics collected from the trace to generate inputs for the stochastic model.

Some of the inputs for the cone model were also derived from the traces, but the cone also needs information on the density of the objects. For this, we used maps based on aerial photography. It allows us to distinguish between buildings, roads and different types of vegetation (trees versus grass; the trace includes a park).   For the area we are studying, the best aerial maps available are from National Agricultural Imagery Program [22]. The most recent photographs are 4-band (RGB plus infrared), with an absolute position accuracy of ~6 m. Therefore, we use a 10m x 10m mask to estimate scatterer density for each position. The accuracy of the classification is limited by image quality and the algorithms. However, this general approach has proven to be effective in practice, and accuracy can be improved with better inputs. Although the processing time for a large area is significant, it can be done off-line and must be performed only once.
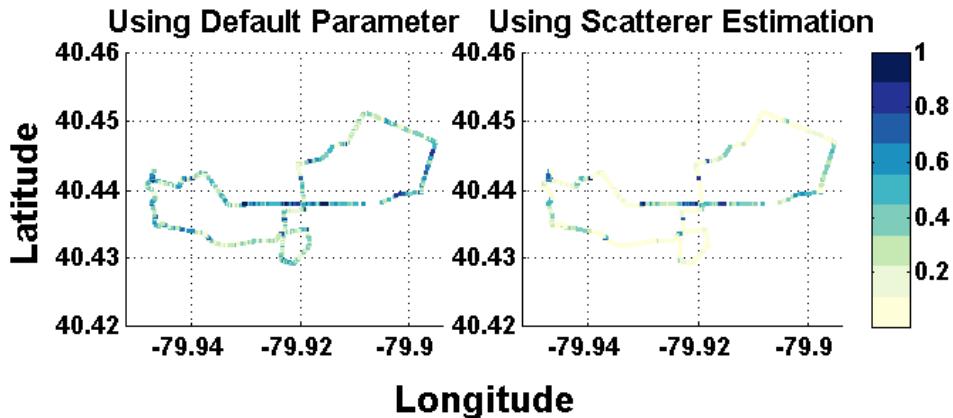
Figure 9: Signal-level Comparison of Spectrum Similarity along Path
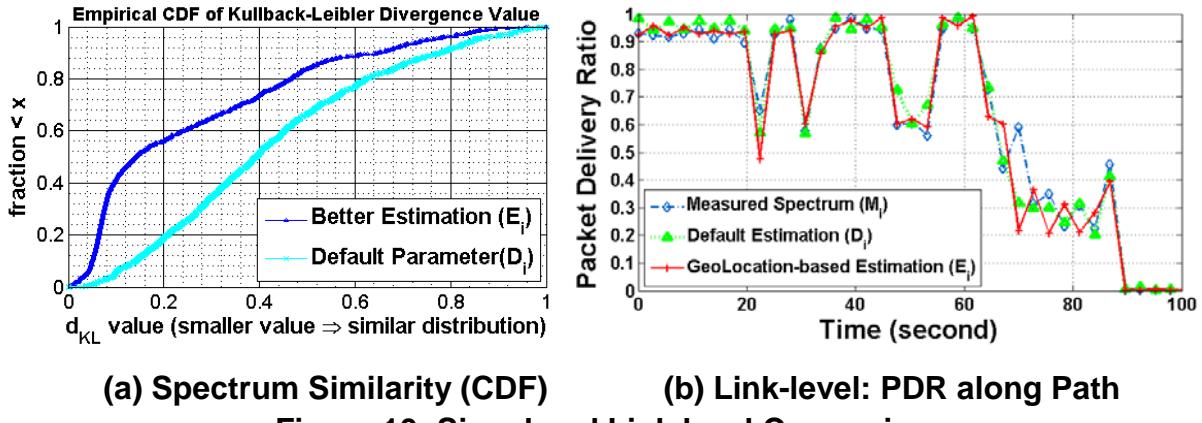
(a) Spectrum Similarity (CDF)   **(b) Link-level: PDR along Path**
Figure 10: Signal and Link level Comparison

Evaluation - We evaluated the accuracy of the models by comparing them with traces, which we used as ground truth, both at the signal level and at the link level. We evaluated the accuracy of the models both for environment-specific input parameters (calculated as described above) and using averaged parameters. The signal-level evaluation compared the Doppler spectrum generated by the models at each point

along the path with that calculated using the trace data. Figure 9 shows the accuracy for both average (left) and environment-specific inputs (righ); lower values mean higher accuracy. We see that the environment-specific parameters improve the accuracy significantly. Figure 10(a) confirms this: it shows the CDFs of the signal level quality metric. Finally, we evaluated the accuracy at the link level by using the emulator to send packets across a wireless link, which was controlled in three ways: using the measured spectrum, and using the models with the averaged (default) and environment-specific inputs. Figure 10(b) shows the packet delivery rate (PDR) as a function of time. We see that using averaged parameters results in a smoother graph while the environment specific inputs track the measurement-based results more closely.

(a) Separate T-R pairs

(b) Common obstacles in T-R pairs

(c) Interleaved T-R pairs

(d) Intersection

(e) Sharing Connectivity Information
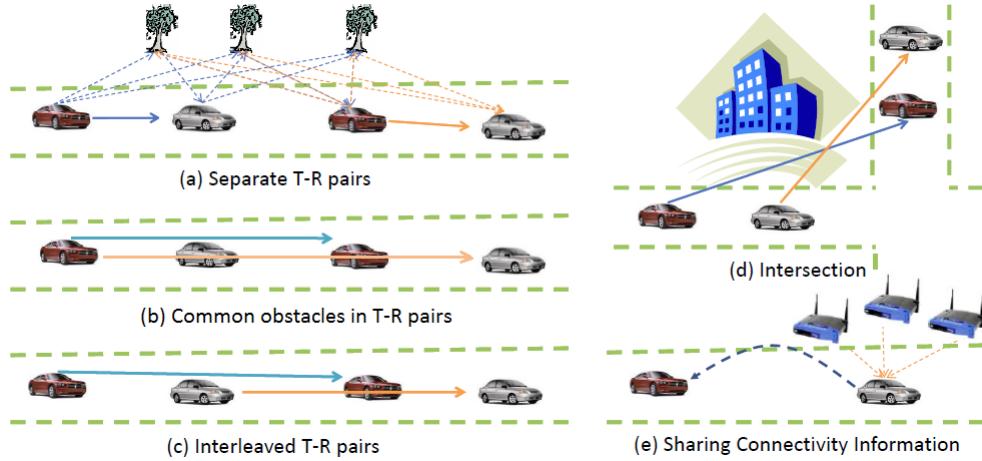
Figure 11: Examples of Spatially Correlated LOS Blocking

4.3.3 Modeling spatial correlation

As explained in Section 1, some protocols exploit spatial correlation between links. This is often also referred to as link diversity. Many wireless simulators and emulators model all links independently, so they assume there is no spatial correlation. This tends to be the best case scenario for protocols that try to exploit link diversity. In practice, since channel properties are determined by the physical environment and nearly links are in very similar environments, we should expect that there often will be some spatial correlation across links. For example, nearby links will be influenced by a similar set of objects with the same of similar properties (density, speed, …) creating similar fading effects, or may have their LOS blocked by the same objects having similar impact on pathloss.

Let us consider LOS blocking in a vehicular environment as an example. Vehicular environments are somewhat unique in the sense that they include, besides large stationary objects (buildings, trees, ..), large mobile objects (trucks, cars, ..). As shown in Figure 11(a)-(d), this results in a number of different examples of spatial correlation for LOS blocking . For example, Figure 11(a) shows two separate transmit-receive pairs; we expect them to experience similar LOS blocking properties due to interleaving cars, but the LOS blocking events will not be synchronized. In contrast, Figure 11(c)

shows interleaved transmit-receive pairs, which will have highly synchronized LOS dynamics. Figure 11(e) shows that we may also need to model spatial correlation consistently over time.

As discussed earlier in the report, channel properties can be modeled using either a geometric or a stochastic approach. For geometric models, the calculated channel properties are completely determined by input (environment details). Assuming consistent information about the environment is used to model all links in the simulation, spatial correlation between links will automatically be captured. In the LOS case, if the same information about the location and size of physical objects (cars, buildings) is used consistently throughout the simulation, all the different examples of spatial correlation shown in Figure 11 will be captured accurately.

A disadvantage of geometric models, however, is that they can be expensive, both to configure and execute. For example in Section 2, we used a signal-level trace to determine LOS blocking for a single wireless link. Signal-level traces are expensive to collect [21] and collecting signal level traces for a large number of links in a coordinated fashion would be very expensive. Stochastic models are often much cheaper since they typically only need averaged information on the physical environment, e.g., the duration and frequency of LOS blocking events. However, stochastic models do not have access to explicit information on correlated physical world impacts. In addition, the independent random process associated with each link introduces additional isolation among multiple channels. In order to have realistic inter-channel correlation, it must be modeled explicitly.

Evaluation - In order to better understand both the complexity of building a separate inter-channel correlation model, and its accuracy compared with a geometric approach, we built a very simple model for LOS blocking correlation [23]. We did not try to model the four scenarios of Figure 11 in detail (this would require four separate models), but instead we built a simple model that enforces a degree of spatial correlation between link pairs. The degree is a function of the distance between the links: nearby links have higher correlation. We compared the geometric and stochastic approach along two dimensions: accuracy and complexity.
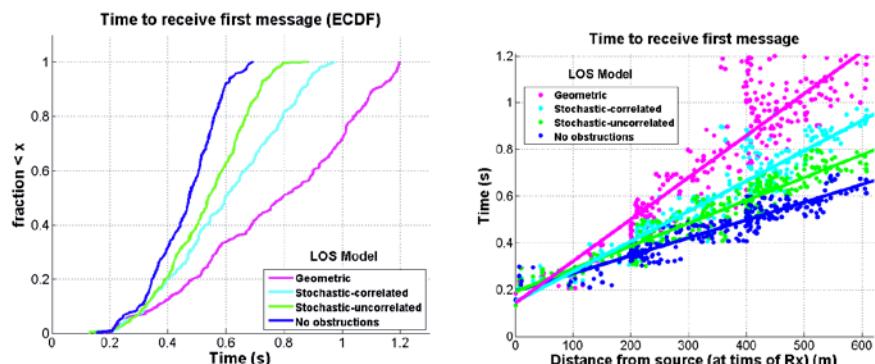


(a) CDF of Delay to Receiver   (b) Delay as a function of Distance
Figure 12: Comparison of Delay for Different Spatial Correlation Approaches

For the evaluation of the precision, we decided to use simulation since it allows us to have a large number of wireless nodes. We implemented support for correlated LOS blocking to ns-3 [24] using both a geometric and a stochastic approach. We used SUMO [25] to generate realistic mobility patterns for a vehicular network operating in a Manhattan grid physical environment. We used a broadcast protocol (e.g., for safety messages) to evaluate how the choice of model impacted the (simulated) performance of an application. The broadcast protocol is based on simple gossip protocol: when a vehicle receives a new message in an incoming packet, it selectively re-broadcasts the new message with a certain probability (to avoid message flooding). Note that, in contrast to Section 2, we do not have ground truth results, so we assume that the geometric approach gives more realistic results since it uses much more detailed information about the physical environment.

Figure 12(a) shows the cumulative distribution function of the delivery times over all nodes in the simulation. In addition to three shadowing models (geometric, and stochastic with and without the explicit spatial correlation modeling), a baseline no obstructions case is included for reference. This shows the performance without any shadowing effects. We picked the parameters for the stochastic models so that the probability of success for an arbitrary link at an arbitrary moment is identical across the geometric and both stochastic models. We observe that the results differ significantly across the models. The "No Obstruction" results have the shortest delay, but, more surprisingly, the results for the correlated geometric and stochastic models also different significantly. Figure 12(b) shows that this is true for all distances. We also see that the results based on the geometric shadowing model have the widest range.

Hop Counts (# of hops)

a. Geometric

b. Stochastic w/ Correlation

c. Stochastic

d. No Obstructions
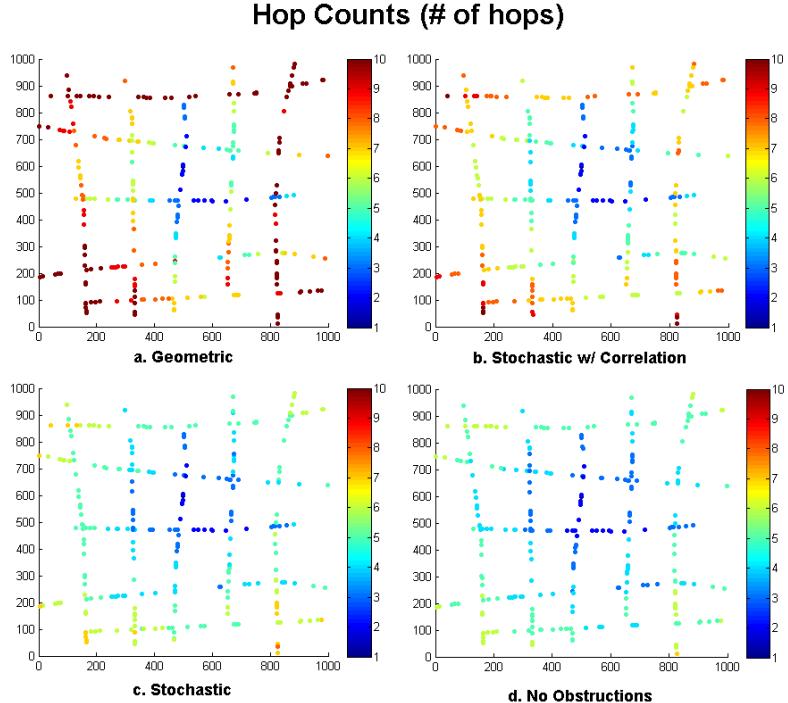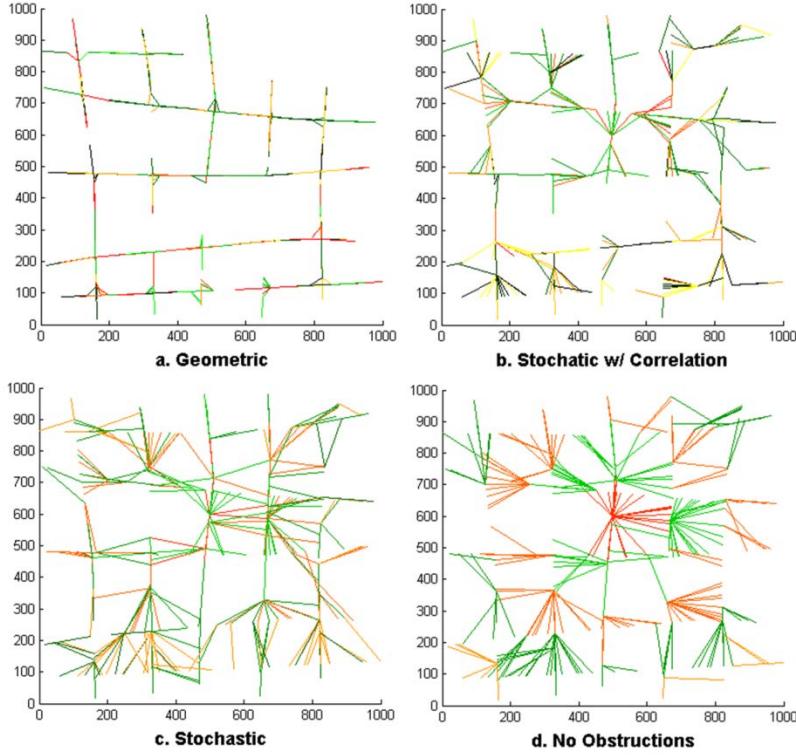
Figure 13: CDF of Delay and (top) Geographic View of the Routes Taken (bottom)

To better understand these results, Figure 13(top) shows the number of hops needed to reach a given destination on a map. Hop count increases with distance for all models, but it increases the fastest for the geometric models and the slowest for the two models that do not enforce spatial correlation. The reason is that the geometric model captures spatial diversity in the most detail, e.g., consistently distinguishing between node pairs on the same road segment, near an intersection, or on parallel roads. The stochastic and no-obstruction models do not make that distinction. This confirmed by Figure 13(bottom), which shows the actual routes taken. With the geometric model, message propagation is mostly confined along the horizontal and vertical road segment, slowing down propagation compared with the uncorrelated models. The simple correlated stochastic model is able to capture this effect to a more limited degree.

We also looked at the complexity of modeling spatial correlation. While stochastic models are relative cheap to execute, the external spatial correlation model (based on a correlation matrix) is expensive. In various modeling steps, the complexity is $O(n_c^4)$ or $O(n_c^6)$, where n_c is the number of wireless nodes whose transmissions/receptions are correlated [23]. In general, it is practical to explicitly model spatial correlation, even in large simulated networks, if for any given link the correlation with most of the links can be ignored (small n_c). Even for small-scale networks with dense links (large n_c), the complexity of calculating correlation increases dramatically, thus an network-wide correlation matrix is more desired.

4.3.4 Modeling external interference

In some simulations it may be necessary to model the impact of external interferers, i.e., transmitters that are not explicitly simulated, on wireless channels. The transmitters may not be simulated because they use a technology not supported by the simulator (e.g., Bluetooth for a WiFi simulation), or for scalability reasons. The primary impact of external transmitters is that they utilize the transmission medium, thus taking away transmission time from simulated transmitters or causing interference for simulated receivers.

Modeling the impact of external transmitters is effectively another example of modeling spatial correlation. The external transmitter will impact the wireless devices in a certain geographic region, effectively creating correlation between a set of links for an "interference" channel feature. The scope of the correlation depends on the nature and range of the interference, which in turn is a function of both the wireless technology used and the transmit range. The results for the LOS blocking case study presented in the previous section should mostly apply since both LOS blocking and the impact of interference are fairly similar.

4.4 Results and Discussions

When the goal is to accurately simulate or emulate wireless networks in environment-specific scenarios, choosing between geometric and stochastic models for specific features is surprisingly complex. A first challenge is that what channel features must be modeled accurately depends on what type of wireless protocols or applications are being evaluated. Features such as fading or spatial correlation may not have much impact in all cases.

With regard to the question of how to model per-link channels features, geometric models have higher accuracy and are preferred. However, collecting the necessary input data can be expensive (e.g., density of objects) while some data may simply not be available (e.g., smaller scatterers). This suggests that a hybrid approach is needed. Use geometric models with accurate inputs based on measurements for the most significant channel features, but stochastic models may have to be used for features that are driven by more fine grain properties of the environment. The model described in [19,26,27] are an example. It uses geometric model for path loss and for fading and LOS blocking created by large objects. However, it uses stochastic models with stochastic inputs for modeling the impact of small objects.

Modeling spatial correlation is relatively simple for features using a geometric model, but it can become very expensive for features that (1) use stochastic models and (2) have correlation across large number of links. There is also a limit to how accurate the models can be, e.g., to distinguish between the scenarios in Figure 11(a)-(d).

5.0 CONCLUSION

Wireless networking has become a part of our everyday life, but it is a very challenging networking technology. The reason is that wireless channel properties, and thus link quality, depend strongly on the details of the physical environment. This creates a significant challenge for both simulation and emulation platforms, since they must accurately model the impact of various physical layer factors on signal propagation in order to achieve a high degree of realism. Moreover, network-level simulation involves many channels so efficiency becomes a major concern.

This project developed a number of techniques to improve the accuracy of developing realistic and efficient environment-specific channel models. First we added a number of features to an emulation-based wireless testbed to improve the accuracy of wireless channel modeling and expand the type of experiments that can be supported. Improvements include better A/D and D/A convertors and a significantly faster control channel that supports more frequent updates of the channel state. We also added large memories to the SCM and DSP card, allowing us to accurately model interference from various types of devices. We also added a front-end that can be tuned across a large spectrum range, opening the door for DSA experiments.

Next, we looked at how the accuracy of both geometric and stochastic channel models can be improved by generating model inputs that accurately reflect a specific environment. We developed a technique that uses aerial photographs to estimate the density of objects in an outdoor environment, which can then be used as inputs for a geometric fading model. We show that using this technique to generate location-specific inputs cuts the modeling error relative to ground truth in half, compared with using average values as inputs. Unfortunately, this technique is quite expensive. This suggests that a hybrid approach is needed. Use geometric models with accurate inputs based on measurements for the most significant channel features (e.g., fading and LOS blocking by large objects, but stochastic models may have to be used for features that are driven by more fine grain properties of the environment (e.g., impact of small objects).

We also look at the problem of modeling spatial correlation between links in the same physical environment, a feature that is often ignored in simulators and emulators. We compare the cost and accuracy of adding spatial correlation to both geometric and stochastic models. Geometric models that use inputs consistently automatically capture spatial correlation, but for stochastic models, an explicit spatial correlation model must be added. While possible, this model becomes expensive as the number of correlated channels increases, so it is not practical for dense wireless networks, or for channel features with a large geographic scope.

Our research suggests a number of additional areas for follow on research. First, support for DSA experiments, both in the form of channel models that are consistent across different parts of the spectrum and control mechanisms for channel switching are needed. Second, our techniques for generate site-specific inputs for geographic models

were demonstrated for a suburban environment and can be extended to other environments. Finally, developing more efficient ways for modeling spatial correlation for stochastic channel models is an important area of research.

References

[1] David Cavin, Yoav Sasson, and Andre Schiper. On the accuracy of manet simulators. In *Workshop on Principles of Mobile Computing (POMC'02)*, pages 38–43. ACM, October 2002.

[2] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions, June 2004. Technical Report TR2004-507, Dept. of Computer Science, Dartmouth College.

[3] Glenn Judd. Repeatable and realistic wireless experimentation through physical emulation, 2006. PhD thesis, Department of Computer Science, Carnegie Mellon University. Also available as Technical report CMU-CS-06-164.

[4] Glenn Judd and Peter Steenkiste. Using Emulation to Understand and Improve Wireless Networks and Applications. In *2nd Symposium on Networked Systems Design and Implementation*. Usenix, May 2005.

[5] Glenn Judd and Peter Steenkiste. Design and Implementation of an RF Front End for Physical Layer Wireless Network Emulation. In *IEEE 2007 Spring Vehicular Technology Conference (VTC)*. IEEE, April 2007.

[6] Glenn Judd and Peter Steenkiste. A software architecture for physical layer wireless network emulation. In *The First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH 2006)*. ACM, September 2006.

[7] Kevin Borries, Xiaohui Wang, Glenn Judd, Peter Steenkiste, and Daniel Stancil. Experience with a wireless network testbed based on signal propagation emulation. In *European Wireless 2010*, April 2010. Invited paper.

[8] Peter Steenkiste. Wireless Emulator Papers, July 2009. http://www.cs.cmu.edu/emulator/papers.html

[9] Glenn Judd and Peter Steenkiste. Characterizing 802.11 Wireless Link Behavior. *Wireless Networks (WINET) Journal*, 2008.

[10] Glenn Judd, Xiaohui Wang, Mei-Hsuan Lu, and Peter Steenkiste. "Using physical layer emulation to optimize and evaluate mobile and wireless systems". In *5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (Mobiquitous 2008)*, July 2008.

[11] Kaushik Lakshminarayanan, Samir Sapra, Srinivasan Seshan, and Peter Steenkiste. Rfdump: An architecture for monitoring the wireless ether. In *CoNEXT '09: Proceedings of the 5th ACM international conference on emerging Networking EXperiments and Technologies*, New York, NY, USA, 2009. ACM.

[12] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In *Proceedings of the ACM SIGCOMM Conference on Network Architectures and Protocols*, Portland, August 2004. ACM.

[13] Mei-Hsuan Lu, Peter Steenkiste, and Tsuhan Chen. On A Time-based Adaptive Retry Strategy for Video Streaming in 802.11 WLANs. *Wireless Communications and Mobile Computing*, 2007.

[14] Glenn Judd and Peter Steenkiste. Understanding Link-level 802.11 Behavior: Replacing Convention with Measurement. In *Wireless Internet Conference 2007 (Wicon07)*, October 2007.

[15] Mei-Hsuan Lu, Peter Steenkiste, and Tsuhan Chen. Design, implementation and evaluation of an efficient opportunistic retransmission protocol. In *MobiCom '09: Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 73–84, New York, NY, USA, 2009. ACM.

[16] Kevin Borries and Daniel Stancil. Efficient Simulation of Mobile-To-Mobile Rayleigh Fading Using Gaussian Quadrature. In *IEEE 2007 Spring Vehicular Technology Conference (VTC)*. IEEE, April 2007.

[17] Jason Matusiak, Richard Graham, Keith Taylor, Eric Anderson, Brenton Walker Using IF-Scale Delays to Emulate the Effects of Site-Specific Multipath in a Digital Wireless Channel Emulator, ACM WiNTECH 2013.

[18] Xiaohui Wang, Kevin Borries, Eric Anderson, Peter Steenkiste, "Network-Scale Emulation of General Wireless Channels", IEEE 74th Vehicular Technology Conference (VTC2011-Fall), September, 2011, San Francisco.

[19] Xiaohui Wang, Eric Anderson, Peter Steenkiste and Fan Bai, "Improving the Accuracy of Environment-specific Vehicular Channel Modeling", The Seventh ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH 2012), held in conjunction with ACM MobiCom 2012, Istanbul, September 2012.

[20] L. Cheng, F. Bai, and D. D. Stancil, "A new geometrical channel model for vehicle-to-vehicle communications," in Proc. IEEE Antennas and Propagation Society Int. Symp. (APS/URSI), 2009.

[21] L. Cheng, B. Henty, D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 ghz dedicated short range communication (dsrc) frequency band," IEEE Journal on Selected Areas in Communications, vol. 25, no. 8, Oct. 2007.

[22] Aerial Photography Field Office, "National agricultural imagery program," http://www.fsa.usda.gov/FSA/apfoapp?area=home&subject=prog&topic=nai, 2010.

[23] Xiaohui Wang, Eric Anderson, Peter Steenkiste and Fan Bai, "Modeling Correlated Wireless Channel Dynamics", conference paper under submission.

[24] The ns-3 network simulator. See http://www.nsnam.org/.

[25] D. Krajzewicz, G. Hertkorn, C. Rossel, and P. Wagner. SUMO (Simulation of Urban Mobility): An Open-Source Traffic Traffic Simulator. In *Middle East Symposium on Simulation and Modeling (MESM)*, 2002.

[26] Xiaohui Wang, Eric Anderson, Peter Steenkiste and Fan Bai, "Improving the Accuracy of Environment-specific Vehicular Channel Modeling", journal submission under review.

[27] Xiaohui Wang, Eric Anderson, Peter Steenkiste, Fan Bai Simulating Spatial Cross-Correlation in Vehicular Networks IEEE Vehicular Networking Conference, 2014.

List of Symbols, Abbreviations, and Acronyms

ADC	Analog Digital Conversion
DAC	Digital Analog Conversion
DSP card	Digital Signal Processing card
SCM	Signal Conversion Module
RFFE	RF Front-End